IN THE CLAIMS

1-23 (Cancelled)

24. (Original) A software architecture for use in a mobile device having a processor and a memory device, comprising:

an application program stored in the memory device and executed by the processor;

a first-order data object stored in the memory device in the form of a first-order data model, wherein the first-order data object includes a second-order data object;

a first-order controller module configured to interface the application program with the first-order data model, wherein the first-order controller module utilizes a generic interface to communicate with the application program; and

a second-order controller module configured to interface the application program with the second-order data object.

25. (Original) The software architecture of claim 24, wherein:

a plurality of application programs are stored in the memory device and executed by the processor;

the first-order controller module is configured to utilize the generic interface to communicate with each of the application programs; and

the second-order controller module is configured to interface the second-order data object with each of the application programs.

26. (Original) The software architecture of claim 24, wherein:

a plurality of first-order data objects of different types are stored in the memory device in the form of first-order data models, wherein each first-order data model corresponds to one type of data object; and

the software architecture further comprising a plurality of first-order controller modules, each first-order controller module being configured to interface the application program with one type of first-order data model, wherein each first-order controller module utilizes the generic interface to communicate with the application program.

27. (Original) The software architecture of claim 24, wherein:

the first-order data object includes a plurality of second-order data objects of different types; and

the software architecture further comprising a plurality of second-order controller modules, each configured to interface the application program with one type of second-order data object.

28. (Original) The software architecture of claim 24, wherein the first-order controller module utilizes a plurality of generic interfaces to communicate with the application program.

29. (Original) The software architecture of claim 24, wherein the second-order controller module is executed from within the first-order controller module.

30. (Original) The software architecture of claim 24, wherein:

the first-order data model is also configured to interact with the second-order data object; and

the second-order controller module is configured to interface the application program with the first-order data model.

31. (Original) The software architecture of claim 30, wherein the second-order controller module utilizes a generic interface to communicate with the application program.

32. (Original) The software architecture of claim 30, wherein the second-order controller module utilizes a generic interface to communicate with the first-order controller module.

33. (Original) The software architecture of claim 24, wherein (1) the second-order data object is stored in the form of a second-order data model, and (2) the second-order controller module is configured to interface the application program with the second-order data model.

34. (Original) The software architecture of claim 33, wherein the second-order controller module utilizes a generic interface to communicate with the application program.

35. (Original) The software architecture of claim 33, wherein the second-order controller module utilizes a generic interface to communicate with the first-order controller module.

36. (Currently Amended) A method of extending a software interface in a mobile device having a plurality of application programs, comprising the steps of:

providing one or more generic interfaces;

providing a plurality of controller modules, each of which utilizes the one or more generic interfaces to communicate with the plurality of application programs; and

providing at least one data model associated with each application program, each data model configured to interface with one of the controller modules;

defining a second-order object within one or more data model;

providing a second-order controller module configured to (1) interface with the data models associated with the second-order object, and (2) utilize the one or more generic interfaces to communicate with the plurality of application programs;

wherein, the second-order controller enables the application programs to interact with the second-order object;

wherein the controller modules enable each application program to interface with each data model.

37. (Original) The method of claim 36, comprising the additional step of:

providing a virtual machine executing on the mobile device, wherein the virtual machine controls the plurality of controller modules and the data models.

38. (Cancelled)

39. (Original) A method of adding functionality to an application program on a mobile device, comprising the steps of:

storing a data object associated with the application program on the mobile device in the form of a data model;

defining one or more second-order objects within the data model;

providing a controller module that interfaces the application program with the data model;

interfacing the application program with the controller module to display the data object;

selecting one of the second-order objects from the displayed data object; and

displaying a list of functions associated with the application program that may be performed on the selected second-order object.

40. (Original) The method of claim 39, wherein the controller module utilizes one or more generic interfaces to communicate with the application program.

41. (Original) The method of 39, wherein the controller module interfaces with the application program to display the list of functions.

42. (Original) The method of claim 39, comprising the additional steps of:

selecting a function from the list of functions; and

executing the function.

43. (Original) The method of claim 39, wherein:

the controller module also interfaces the data model with one or more other application programs on the mobile device; and

the list of functions includes one or more functions that are associated with one of the other application programs.

44. (Original) The method of claim 43, comprising the additional steps of:

selecting a function from the list of functions;

determining whether the selected function is associated with the application program or one of the other application programs;

if the selected function is associated with the application program, then executing the function using the application program; and if the selected function is associated with one of the other application programs, then (1) launching the other application program and (2) executing the function using the other application program.

45. (Original) The method of claim 44, wherein the controller module performs the step of determining whether the selected function is associated with the application program or one of the other application programs.

46. (Original) The method of claim 44, wherein the controller module launches the other application program.

47. (Original) The method of claim 39, comprising the additional steps of:

storing the one or more second-order objects in the form a second-order data models; and

providing a second-order controller module that interfaces the application program with the second-order data models.

48. (Original) The method of claim 47, wherein the second-order controller module utilizes one or more generic interfaces to communicate with the application program.

49. (Original) The method of claim 47, wherein the second-order controller module is implemented by the controller module when one of the second-order objects is selected from the displayed data object.

50. (Original) The method of claim 47, wherein the second-order controller module interfaces with the application program to display the list of functions.

51. (Original) The method of claim 39, comprising the additional step of:

storing the one or more second-order data objects in the form of second-order data models; and

providing a second-order controller module that interfaces the controller module with the second-order data models.

52. (Original) The method of claim 51, wherein the second-order controller module utilizes one or more generic interfaces to communicate with the controller module.

53. (Original) The method of claim 51, wherein the second-order controller module is implemented by the controller module when one of the second-order objects is selected from the displayed data object.

54. (Original) The method of claim 39, comprising the additional step of:

providing a second-order controller module that interfaces the application program with the one or more second-order objects using the data model.

55. (Original) The method of claim 54, wherein the second-order controller module utilizes one or more generic interfaces to communicate with the application program.

56. (Original) The method of claim 54, wherein the second-order controller module is implemented by the controller module when one of the second-order objects is selected from the displayed data object.

57. (Original) The method of claim 54, wherein the second-order controller module interfaces with the application program to display the list of functions.

58. (Original) The method of claim 39, comprising the additional step of:

providing a second-order controller module that interfaces the controller module with the one or more second-order objects using the data model.

59. (Currently Amended) The method of claim 58, wherein the second-order controller module utilizes one or more generic ~~interface~~ interfaces to communicate with the controller module.

60. (Original) The method of claim 58, wherein the second-order controller module is implemented by the controller module when one of the second-order objects is selected from the displayed data object.

61. (Original) A method of adding functionality to an e-mail messaging application, comprising the steps of:

storing an e-mail message data object in the form of an e-mail data model; defining one or more second-order objects within the e-mail message data object;

providing a first-order controller module that interfaces the e-mail messaging application with the e-mail message data object through the e-mail data model;

providing a second-order controller module that interfaces the e-mail messaging application with the second-order objects through the e-mail data model;

interfacing the e-mail message data object with the first-order controller module to display the e-mail message;

selecting one of the second-order objects from within the displayed e-mail message data object;

interfacing the second-order controller module with the e-mail messaging application to display a list of functions associated with the e-mail messaging application that may be performed on the selected second-order object;

selecting a function from the displayed list of functions; and

executing the function.

62. (Original) The method of claim 61, wherein:

the first and second order controller modules also interface with one or more application programs other than the e-mail messaging application; and

the displayed list of functions includes one or more functions that are associated with one of the application programs other than the e-mail messaging application.

63. (Original) The method of claim 62, comprising the additional steps of:

determining whether the selected function is associated with the e-mail messaging application or one of the other application programs;

if the selected function is associated with the e-mail messaging application, then executing the function using the e-mail messaging application; and

if the selected function is associated with one of the other application programs, then (1) launching the other application program and (2) executing the function using the other application program.

64. (Original) The method of claim 61, wherein the one or more second-order objects are addresses within e-mail message data object.

65. (Original) The method of claim 61, wherein:

the one or more second-order objects are defined by a third-party; and

the second-order controller is designed by the third-party and installed by a user to upgrade the e-mail messaging application.

66. (New) The software architecture of claim 24, further comprising:

a virtual machine stored in the memory device and executed by the processor, wherein the virtual machine executes each controller module and corresponding data model.

67. (New) The software architecture of claim 66, wherein the virtual machine is an object oriented run-time environment.

68. (New) The software architecture of claim 66, further comprising:

an operating system stored in the memory device and executed by the processor, wherein the virtual machine is executed by the operating system.

69. (New) The software architecture of claim 24, wherein the first order controller module is configured to interface the application programs with an e-mail message data model.

70. (New) The software architecture of claim 24, wherein the first order controller module is configured to interface the application programs with an address book card data model.

71. (New) The software architecture of claim 24, wherein the first order controller module is configured to interface the application programs with a calendar entry data model.

72. (New) The software architecture of claim 24, wherein the first order controller module is configured to interface the application programs with a task list data model.

73. (New) The software architecture of claim 24, wherein the first order controller module is configured to interface the application programs with a memo data model.

74. (New) The software architecture of claim 24, wherein the generic interface enables the first order controller to interface with an application program installed on the mobile device that supports the generic interface.

75. (New) The software architecture of claim 24, wherein the generic interface is configured to perform an operation with any arbitrary data model.

76. (New) The software architecture of claim 24, wherein the application program is configured to query the first-order controller module to determine whether the first-order controller module supports a particular type of generic interface.

77. (New) The software architecture of claim 24, wherein additional controller modules are added to the software architecture that support one or more additional generic interfaces.

78. (New) The software architecture of claim 24, wherein the generic interface includes a paint provider interface for painting the data objects into the application program.

79. (New) The software architecture of claim 24, wherein the generic interface includes a field provider interface for providing the application program with one or more fields from the data model.

80. (New) The software architecture of claim 24, wherein the generic interface includes a verb provider interface for providing the application program with one or more functions specific to the data model.

81. (New) The software architecture of claim 24, wherein the data objects are logged in a persisted list when stored in the memory device, and the persisted list identifies the data model corresponding to each data object.

82. (New) The software architecture of claim 24, wherein only one instance of each controller module is executing at one time.

83. (New) The software architecture of claim 82, wherein the controller module associated with a particular type of data model is identified to one of the plurality of application programs by the data model when the application program attempts to interact with the data model.

84. (New) The method of claim 39, comprising the additional step of:

providing a virtual machine that executes each controller module and corresponding data model.

85. (New) The method of claim 84, wherein the virtual machine is an object oriented run-time environment.

86. (New) The method of claim 39, comprising the additional step of:

providing an operating system stored in the memory device and executed by the processor, wherein the virtual machine is executed by the operating system.

87. (New) The method of claim 39, wherein the controller module is configured to interface the application programs with an e-mail message data model.

88. (New) The method of claim 39, wherein the controller module is configured to interface the application programs with an address book card data model.

89. (New) The method of claim 39, wherein the controller module is configured to interface the application programs with a calendar entry data model.

90. (New) The method of claim 39, wherein the controller module is configured to interface the application programs with a task list data model.

91. (New) The method of claim 39, wherein the controller module is configured to interface the application programs with a memo data model.

92. (New) The method of claim 39, wherein the generic interface enables the controller to interface with an application program installed on the mobile device that supports the generic interface.

93. (New) The method of claim 39, wherein the generic interface is configured to perform an operation with any arbitrary data model.

94. (New) The method of claim 39, wherein the application program is configured to query the controller module to determine whether the controller module supports a particular type of generic interface.

95. (New) The method of claim 39, wherein additional controller modules are added to the software architecture that support one or more additional generic interfaces.

96. (New) The method of claim 39, wherein the generic interface includes a paint provider interface for painting the data objects into the application program.

97. (New) The method of claim 39, wherein the generic interface includes a field provider interface for providing the application program with one or more fields from the data model.

98. (New) The method of claim 39, wherein the generic interface includes a verb provider interface for providing the application program with one or more functions specific to the data model.

99. (New) The method of claim 39 further comprising the step of:

storing data objects in a persisted list, wherein the persisted list identifies the data model corresponding to each data object.

100. (New) The method of claim 95, wherein only one instance of each controller module is executing at one time.

101. (New) The method of claim 100, wherein the controller module associated with a particular type of data model is identified to one of the plurality of application programs by the data model when the application program attempts to interact with the data model.